1. Terms and Definitions

- A rule $r$ is an expression of the form
  \[ h \leftarrow b_1, \ldots, b_n, \neg c_1, \ldots, \neg c_m. \]

- We use the following notations for a rule $r$:
  \[
  H(r) = h \quad \text{(head)}
  \]
  \[
  B(r) = \{b_1, \ldots, b_n, \neg c_1, \ldots, \neg c_m\} \quad \text{(body)}
  \]
  \[
  B^+(r) = \{b_1, \ldots, b_n\}
  \]
  \[
  B^-(r) = \{c_1, \ldots, c_m\}
  \]

- We define normal (logic) programs as sets of rules.

2. Syntactic Restrictions

- We distinguish the following special cases:
  - positive rules: $h \leftarrow b_1, \ldots, b_n$
  - atomic rules: $h \leftarrow \neg c_1, \ldots, \neg c_m$
  - strictly unary rules: $h \leftarrow b, \neg c_1, \ldots, \neg c_m$
  - strictly binary rules: $h \leftarrow b_1, b_2, \neg c_1, \ldots, \neg c_m$

- We extend these conditions for sets of rules:
  - positive programs: $\forall r \in P : |B^-(r)| = 0$
  - atomic programs: $\forall r \in P : |B^+(r)| = 0$
  - unary programs: $\forall r \in P : |B^+(r)| \leq 1$
  - binary programs: $\forall r \in P : |B^+(r)| \leq 2$
**Least Models**

If $P$ is a positive normal program, then

1. $P$ has a unique minimal model, i.e. the least model $LM(P)$ of $P$;
2. $LM(P) = T_P \uparrow \omega = lfp(T_P)$ where the immediately true operator $T_P$ is defined for all $A \subseteq HB(P)$ by $T_P(A) = \{H(r) \mid r \in P \text{ and } B^+(r) \subseteq A\}$;
3. and $lfp(T_P) = T_P \uparrow i$ for some $i \in \mathbb{N}$, if $P$ is finite.

---

**Stable and Supported Models**

**Definition.** Given an interpretation $M$, the Gelfond-Lifschitz reduct $P^M = \{r^+ \mid r \in P \text{ and } B^-(r) \cap M = \emptyset\}$

where $r^+$ is defined as $H(r) \leftarrow B^+(r)$ for $r \in P$.

**Definition.** For a normal program $P$, an interpretation $M \subseteq HB(P)$ is

1. a stable model of $P$ $\iff$ $M = LM(P^M)$, and
2. a supported model of $P$ $\iff$ $M = T_{pol}(M)$.

---

**Level Numbers**

**Definition.** For each atom $b \in LM(P)$, the level number $\text{lev}(b)$ of $b$ is the least number $n$ such that $b \in T_P \uparrow n - T_P \uparrow (n - 1)$.

**Example.** Consider a positive normal program $P = \{r_1 \leftarrow a; \ r_2 = a \leftarrow b; \ r_3 = b \leftarrow a\}$ with $LM(P) = \{a, b\}$ and the corresponding level numbers $\text{lev}(a) = 1$ and $\text{lev}(b) = 2$.

---

**Simple Example**

**Example.** The normal program $P = \{a \leftarrow b; \ b \leftarrow a\}$ has two supported models $M_1 = \emptyset$ and $M_2 = \{a, b\}$.

However, only $M_1$ is stable, as

1. $LM(P^{M_1}) = LM(P) = \emptyset = M_1$ and
2. $LM(P^{M_2}) = LM(P) = \emptyset \neq M_2$. 
Properties of Stable and Supported Models

1. Stable models are also supported models.
2. Stable and supported models coincide for atomic programs.
3. The classical models of the completed program (as proposed by Clark in 1978) correspond to supported models.

Definition. Clark’s completion. For each $a \in HB(P)$ and the rules $r_1, \ldots, r_d \in P$ with $H(r_i) = a$ are translated into

$$a \leftrightarrow (T_{Clark}(B(r_1))) \lor \ldots \lor (T_{Clark}(B(r_d)))$$

where $T_{Clark}(B(r_i)) = b_1 \land \cdots \land b_n \land \neg c_1 \land \cdots \land \neg c_m$ for the body $B(r_i) = \{b_1, \ldots, b_n, \text{not } c_1, \ldots, \text{not } c_m\}$ of $r_i$.

Capturing Stable Models

Let $M$ be a supported model of a normal logic program $P$.

Proposition. If $#$ is a level numbering w.r.t. $M$, then it is unique.

Theorem. $M$ is a stable model of $P$ if and only if there is a level numbering $#$ w.r.t. $M$.

2. Characterizing Stability

Definition. Let $M$ be a supported model of a normal program $P$. A level numbering w.r.t. $M$ is

a function $#: M \cup SR(P, M) \rightarrow \mathbb{N}$ such that

1. for all $a \in M$, $#a = \min\{#r \mid r \in SR(P, M) \text{ and } a = H(r)\}$ and
2. for all $r \in SR(P, M)$, $#r = \max\{#b \mid b \in B^+(r)\} + 1$

where $SR(P, M) = \{r \in P \mid M \models B(r)\}$.

We define $\max \emptyset = 0$ to cover rules $r$ with $B^+(r) = \emptyset$.

Example. Recall the supported models of $P = \{r_1, r_2\}$ with $r_1 = a \leftarrow b$ and $r_2 = b \leftarrow a$: $M_1 = \emptyset$ and $M_2 = \{a, b\}$.

- Since $M_1 \cup SR(P, M_1) = \emptyset$, $M_1$ is trivially stable.
- For $M_2$, the domain $M_2 \cup SR(P, M_2) = M_2 \cup P$ and the resulting set of equations

$$#a = #r_1, \quad #r_1 = #b + 1,$$
$$#b = #r_2, \quad #r_2 = #a + 1$$

has no solution. Thus $M_2$ is not stable.
3. Clausal Representation

- We use an atomic normal program $\text{Tr}_{AT}(P) = \text{Tr}_{SUPP}(P) \cup \text{Tr}_{CTR}(P) \cup \text{Tr}_{MIN}(P) \cup \text{Tr}_{MAX}(P)$ as an intermediary representation when translating a normal program $P$ into a set of clauses $\text{Tr}_{CL}(\text{Tr}_{AT}(P))$.
- Level numbers have to be captured using binary counters which are represented by vectors of propositional atoms.
- Certain primitives are formalized as subprograms: $\text{SEL}(c)$, $\text{NXT}(c,d)$, $\text{FIX}(c,d)$, $\text{LT}(c,d)$, and $\text{EQ}(c,d)$.

Optimizations

- The level numbers associated with rules can be totally omitted, if all non-binary rules $r$ with $|B^+(r)| > 2$ are translated away.
- A normal logic program $P$ is partitioned into its strongly connected components $C_1, \ldots, C_n$ on the basis of positive dependencies.
- No counters are needed, if $|H(C_i)| = 1$ holds.
- The number of bits $\nu C_i = \lfloor \log_2(|H(C_i)| + 2) \rfloor$ for other strongly connected components $C_i$.
- Fixed translation schemes can be devised for atomic, strictly unary, and strictly binary rules.

Example

For $P = \{ a \leftarrow b; \, b \leftarrow a \}$, the translation $\text{Tr}_{AT}(P)$ contains the following rules for $a$:

$b \leftarrow \neg \text{bt}(r_2); \; \text{bt}(r_2) \leftarrow \neg \text{bt}(r_2); \; \text{bt}(r_2) \leftarrow \neg \mathcal{A};$

$\mathcal{A} \leftarrow \neg a; \; \times \leftarrow \neg x, \neg \mathcal{A}, \neg \text{min}(a);$  

$x \leftarrow \neg x, \neg \text{bt}(r_2), \neg \text{bt}(nxt(a), \text{ctr}(b));$  

and $\text{min}(b) \leftarrow \neg \text{bt}(r_2), \neg \text{eq}(\text{nxt}(a), \text{ctr}(b))$

in addition to four subprograms for choosing the values of ctr(a) and nxt(a) as well as comparing the latter with ctr(b). Rules that have to be introduced for $b$ are symmetric.

The only stable model is $N = \{ \mathcal{A}, \mathcal{B}, \text{bt}(r_1), \text{bt}(r_2) \}$.

4. Experiments

- We have implemented $\text{Tr}_{AT}$ and $\text{Tr}_{CL}$ as respective translators $\text{LP2ATOMIC}$ and $\text{LP2SAT}$ to be used together with $\text{LPARSE}$.
- Our experiments were run on a 1.67 GHz CPU with 1GB memory.
- In our benchmark, we compute all subgraphs of $D_n$ whose all vertices are mutually reachable.

Here $D_n = \langle V_n, E_n \rangle$ is a directed graph with $n$ vertices and $n^2 - n$ edges:

$V_n = \{ 1, \ldots, n \}$ and  

$E_n = \{ (i,j) \mid 0 < i \leq n, 0 < j \leq n, i \neq j \}$. 
Our benchmark problem is formalized as follows:

\begin{verbatim}
vertex(1..n).
in(V1,V2) :- not out(V1,V2), vertex(V1;V2), V1!=V2.
out(V1,V2) :- not in(V1,V2), vertex(V1;V2), V1!=V2.
reach(V,V) :- vertex(V).
reach(V1,V3) :- in(V1,V2), reach(V2,V3), vertex(V1;V2;V3), V1!=V2, V1!=V3.
\end{verbatim}

The order in which the reachability of nodes inferred cannot be determined beforehand.

---

### Computing Only One Solution

<table>
<thead>
<tr>
<th>Number of Vertices</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMODELS</td>
<td>0.009</td>
<td>0.013</td>
<td>0.022</td>
</tr>
<tr>
<td>CMODELS</td>
<td>0.046</td>
<td>0.042</td>
<td>0.055</td>
</tr>
<tr>
<td>LP2ATOMIC+SMODELS</td>
<td>$&gt;10^4$</td>
<td>$&gt;10^4$</td>
<td>$&gt;10^4$</td>
</tr>
<tr>
<td>LP2SAT+CHAFF</td>
<td>0.771</td>
<td>32.6</td>
<td>254</td>
</tr>
<tr>
<td>LP2SAT+RELSAT</td>
<td>2.51</td>
<td>$&gt;10^4$</td>
<td>$&gt;10^4$</td>
</tr>
<tr>
<td>WF+LP2SAT+RELSAT</td>
<td>2.80</td>
<td>4830</td>
<td>$&gt;10^4$</td>
</tr>
<tr>
<td>ASSAT</td>
<td>0.023</td>
<td>0.028</td>
<td>0.037</td>
</tr>
</tbody>
</table>

---

### Computing All Solutions

<table>
<thead>
<tr>
<th>Number of Vertices</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMODELS</td>
<td>0.004</td>
<td>0.003</td>
<td>0.003</td>
<td>0.033</td>
<td>12</td>
</tr>
<tr>
<td>CMODELS</td>
<td>0.031</td>
<td>0.030</td>
<td>0.124</td>
<td>293</td>
<td>-</td>
</tr>
<tr>
<td>LP2ATOMIC+SMODELS</td>
<td>0.004</td>
<td>0.008</td>
<td>0.013</td>
<td>0.393</td>
<td>353</td>
</tr>
<tr>
<td>LP2SAT+CHAFF</td>
<td>0.011</td>
<td>0.009</td>
<td>0.023</td>
<td>1.670</td>
<td>-</td>
</tr>
<tr>
<td>LP2SAT+RELSAT</td>
<td>0.004</td>
<td>0.005</td>
<td>0.018</td>
<td>0.657</td>
<td>1879</td>
</tr>
<tr>
<td>WF+LP2SAT+RELSAT</td>
<td>0.009</td>
<td>0.013</td>
<td>0.018</td>
<td>0.562</td>
<td>1598</td>
</tr>
<tr>
<td>Models</td>
<td>1</td>
<td>1</td>
<td>18</td>
<td>1606</td>
<td>565080</td>
</tr>
<tr>
<td>SCCs with $</td>
<td>H(C)</td>
<td>&gt; 1$</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Rules (LPARSE)</td>
<td>3</td>
<td>14</td>
<td>39</td>
<td>84</td>
<td>155</td>
</tr>
<tr>
<td>Rules (LP2ATOMIC)</td>
<td>3</td>
<td>18</td>
<td>240</td>
<td>664</td>
<td>1920</td>
</tr>
<tr>
<td>Clauses (LP2SAT)</td>
<td>4</td>
<td>36</td>
<td>818</td>
<td>2386</td>
<td>7642</td>
</tr>
<tr>
<td>Clauses (WF+LP2SAT)</td>
<td>2</td>
<td>10</td>
<td>553</td>
<td>1677</td>
<td>5971</td>
</tr>
</tbody>
</table>

---

### 5. Discussion

- The new characterization of stable models is based on canonical level numberings of atoms and rules.
- The translation function $\mathit{Tr}_\mathit{AT} \circ \mathit{Tr}_\mathit{CL}$ has distinctive properties:
  1. it covers all finite normal programs $P$;
  2. a bijective relationship of models is obtained,
  3. the Herbrand base $\mathit{HB}(P)$ is preserved,
  4. the length $||\mathit{Tr}_\mathit{CL}(\mathit{Tr}_\mathit{AT}(P))||$ is of order $||P|| \times \log_2 |\mathit{HB}(P)|$, and
  5. incremental updating is not needed.
Conclusions and Future Work

- Various kinds of closures of relations, such as transitive closure, can be properly captured with classical models.
- Our approach is competitive against other SAT-solver-based approaches when the task is to compute all stable models.
- Further optimizations should be pursued for in order to really compete with SMODELS.
- In the future, we intend to study techniques to reduce the number of binary counters and the numbers of bits involved in them.